Change log

v1.07.1 2024-08-09

Changed

- 64-bit `makefile` now supports native/default macOS `Clang` compiler
- `AuxInfo`s in CLI version of `InChI` are now identical to `v1.06`
- Intel oneAPI DPC++/C++ compiler can now be used with Microsoft `Visual Studio` solutions/projects on Microsoft Windows
- "Replace memchr() check with a new function to check the element group" pull request merged
- "Faster Element Symbol Handling" pull request merged
- "replace all instances of old license text. Fixes #33" pull request merged

Fixed

- "Issues with InChI I/O with /FixedH / Issue #27 and "Seqmentation fault in inchi-1.exe with /InChI2InChI validation / Issue #28 have been fixed
- Addressing Google oss-fuzz issues
- Minor bugs fixes

v1.07.0 2024-07-17

Changed

- 9 mathematical functions had to be rewritten in `ichister.c` and `e_0dstereo.c` to address functions' arguments issues related to arrays of various dimensions
- 25 blocks of code in files in `runichi2.c`, `inchi_dll.c`, `inchi_dll_a2.c`, `ichiprt3.c`, `ichiread.c`, `ichirvr1.c`, `runichi.c`, `ichiparm.c`, `ichitaut.c` had to be rewritten to address:
  - memory leaks
  - security issues
  - buffer overruns
  - improperly written conditional statements or bit-wise operations
- makefile/makefile32 files and `Microsoft® Visual Studio` projects/solutions have been updated and revised
- `__isascii/isascii` macro support provided for all compilers

Fixed (security)

- 33 buffer overflow issues due to use of large array dimensions
- 157 security bugs related to improper `NULL` pointer dereferencing which might cause crashes or exits
- 71 memory leaks
- 530 potential applications of optional bounds checking functions

Additionally, 29 potential security issues have been marked for further revision.

Fixed

- 2480 bugs and issues have been addressed:
    - type conversions and mismatches
    - removing redundant variables and/or code
    - addressing `Clang/LLVM` warnings

v1.06 2020-12-20

Added

- Zz pseudoelement atoms both in polymeric and general-placeholder context Option `NPZZ` to allow non-polymer Zz atoms (polymeric ones are allowed by default by Polymers switch)
- Option `Polymers105` to emulate v. 1.05 treatment of polymers
- Support of simultaneous presence of both source-based and structure-based subunits in copolymer unit (per request of Gerd Blanke, StructurePendium)
- Option `NoFrameShift` (disables frame shift for all CRU's), works for both inchi-1 and API
- In-CRU inner repeats are performed if the switch `FoldCRU` is specified (so *-(CH2CH2)n-* is converted to *-(CH2)n-* and so on)
- Switch `NoEDits` disables both frame shift and CRU folding
- Option `SAtZz` enables stereo at atoms connected to Zz pseudo atoms (default: disabled/ignored)
- Switch `LooseTSACheck` relaxes strictness of tetrahedral stereo ambiguity check for stereo atoms in cycles (useful for dealing with large cycles 'cleaned' by some software), per request from FDA
- Switch `WMnumber` for InChI Library/inchi-1 (sets timeout in milliseconds, strictly requires long int number), per request from CDK folks
- `NoWarnings` option to inchi-1 and InChI Library: suppress warnings but keep error messages; useful for the long runs on multi-million record inputs
- Switch `MergeHash`: make combined InChIKey+extra hash(es) if present (inchi-1)
- Switch `PERTHREAD`:n allowing one to process n SDF records in each of mol2inchi threads.
- Use Tab as synonym to Tabbed, works for inchi-1
- API call `IXA_INCHIBUILDER_SetOption_Timeout_Milliseconds()`
- `IXA_MOL_GetBondOtherAtom()` exposed in the public API, as Paul Thiessen suggested
- API call `IXA_MOL_ReserveSpace()` (necessary in new mode `IXA_USES_SMART_ALLOCS`)
- More IXA API functionality for polymers (API calls `IXA_MOL_CreatePolymerUnit()`, `IXA_MOL_SetPolymerUnit()`, `IXA_MOL_GetPolymerUnitId()`, `IXA_MOL_GetPolymerUnitIndex()`)
- Made /inchi2inchi calc mode to treat InChI=1//

- Provided a simplistic example of multi-threaded boss-worker InChI generation into mol2inchi demo program (Linux pthreads/Windows threads are used)
- winchi-1: added SDF navigation by keyboard Ctrl-Lt, Ctrl-Rt arrows; Ctrl-G for Goto Compound #

Changed

- Polymers with undefined ("star", or Zz) end-groups are represented with explicitly shown in InChI string Zz atoms. If CRU frame shift took place, bonding is changed accordingly
- In InChI part for any polymer CRU , the crossing bond printed first is now that pointing to more senior CRU end ("head")
- Returned value of `IXA_INCHIBUILDER_OPTION_LargeMolecules` to 11 as it was in v1.05 but 31 in v1.06pre3 (per request by Daniel Lowe)
- Improved performance of IXA memory allocations by using expandable arrays (per request by Daniel Lowe; see `#ifdef IXA_USES_SMART_ALLOCS`)
- Changed inchi-1 SDF ID value type to unsigned long to account for current CAS numbers which may now be > LONG MAX (thanks to DT for noticing)
- Made return codes of `GetInChIFrom...()` functions in case of InChI read error corresponding to what is declared in API description
- Changed `INFINITY` defined in chi_can2.c to `INCHI_CANON_INFINITY` to avoid collision with MS' INFINITY defined in VS 2015 math.h (pointed out by DT)
- Changed label of winchi-1 toolbar button "Write result" to more descriptive "Batch process all"
- Changed winchi-1 disk output behavior to avoid unnecessary writing, per ACD/Labs request . Now the program writes log/output/problem files disk only in batch mode, if not otherwise requested
- Now winchi-1 starts in maximized window.
- Made numerous small changes related to the refactoring of code

Removed

- Removed function char `base26_checksum()` (as there is no check character in InChIKey anymore)
- Removed InChI2InChI test option of inchi-1 executable (use RTrip option of test_ixa demo app instead, or use API directly)

Fixed (security)

- Fixed 22 possibly security-compromising issues found by Google-AutoFuzz of 2018-2019 (reported via SourceForge) mostly issues related to inadequate handling of invalid/artificial input on inchi2struct conversion that may result in memory corruption, etc.

- Fixed 102 more, then 1 more, possibly security-compromising issues found by Google- AutoFuzz of 2019 (reported by Ian Wetherbee), mostly issues related to inadequate handling of invalid/artificial input on inchi2struct conversion that may result in memory corruption, etc.; some are similar to/same as GAF22
- Fixed 19 more possibly security-compromising issues of 2020 found by google/oss-fuzz
- Fixed 4 possibly security-compromising issues of 2019 (reported by Cure53) issues found with AddressSanitizer, mostly related to inadequate handling of invalid/artificial input that may result in memory corruption, etc.

Fixed (other)

- Fixed normalization bug resulting in change of InChI string upon atom renumbering for some molecules having acidic hydroxy group at cationic heteroatom center. Total of ~1700 PubChem Compound entries out of ~102M are favorably affected, see KNOWNISSUES.md section on 1.05 issues. In particular, this fix repairs the issue reported by Lutz Weber, Ontochem, on 2020-03-03.
- Fixed bug resulting in InChI Error -30010 (STEREOCOUNT_ERR) – which appeared on CIDs 124897603, 124921144 as was found on massive testing with PubChem dataset. "The failure occurs when one of two or more constitutionally equivalent undefined stereocenters has been removed due to stereo equivalence of its two attachments." Thanks, DT!
- Fixed bug reported by Andrew Dalke and separately by Burt Leland: H isotope with insane mass difference has been silently consumed which sometimes resulted in memory corruption
- Fixed bug of not recognizing spiro chirality like in olean, described in: Maeda et al., Biophys.Physicobiol., 2018, 87 (thanks to DT)
- Added check to fix bug with NULL szXtra in ikey_dll.c, thanks to Wolf-Dietrich Ihlenfeldt
- Fixed "data race" bug(s) reported by John Salmon in inchi-discuss and similar ones (by adding el_number constants instead of static variables)
- Fixed critical race condition bug pointed out by Karl Nedwed, strutil.c
- Fixed bug in polymer CRU canonicalization in inchi2struct mode at comparing seniority of 'junior members' for various possible pairs of 'star' atom attachment points
- Fixed bug in inchi2struct for polymers exhibited at multipliers in the formula
- Fixed issue in polymer treatment: enabled CRU frame (phase) shift only for "head-to-tail" connection; fixed several bugs for star-ended CRU's sometimes resulting in crash
- Fixed bug which caused crash on read V3000 file alkane-c4000.mol
- Fixed bug "for V3000 AuxInfo misses coordinates /rC:" (tested: methane.sdf, pdb-1000-to-1023-atoms.sdf).
- Fixed several bugs in AuxInfo output (some of general nature and some

specifically of IXA)

- Fixed the case of erroneous behavior of inchi-1 with "/D" option Added treating of mistakenly overlooked case of zero bond number in MOL V3000 parser (thanks to Burt Leland)
- Fixed typo in ichirvr2.c#2838 and removed junk UTF symbols appeared due to copy-pasting from PDF
- Fixed minor issues with output in /Tabbed mode
- Fixed: multi-threaded mol2inchi did not treat MOL files (SDF was OK)
- Fixed issue of winchi-1 making no output to result's text window after calling "Process all" from record !=1
- Fixed issue of unnecessarily repeating some output lines in winchi-1 InChI/AuxInfo/annotation window

v1.05 release 2017-02-04

Added

- Support for chemical element numbers 113-118 (see release notes for detailed information)
- Experimental support of InChI/InChIKey for regular single-strand polymers (see release notes for detailed information)Several minor bugfixes is necessary for treating large molecules (previous versions supported only V2000 format limited to not more than 1000 atoms).
- Provisional support for extended features of Molfile V3000
- API procedure for direct conversion of Molfile input to InChI, as well as demo program illustrating its use
- New set of API procedures for both low and high-level operations (InChI extensible interface, IXA), as well as a demo program illustrating its use
- Convience options added to inchi-1 executable
- New optin `Large Molecules` instructs inchi-1 executable to accept molecules containing more than 1024 (but less than 32767) atoms
- A new option `Polymers` instructs inchi-1 executable to accept polymer data in input V2000 Molfiles
- The new option "Output at Error an empty InChI", `OutErrInChI` (`/OutErrInChI` under Windows, `-OutErrInChI` under Linux) instructs inchi-1 executable to output empty InChI and corresponding InChIKey if error occurs (default behaviour is output nothing)
- Option `Record:N` (`/Record:N` under Windows, `-Record:N` under Linux) instructs inchi-1 executable to process only the N-th record of the input file in SDF format

Changed

- Implementation of wildcard expansion in "Allow Multiple Input (AMI) mode" (like in `inchi-1 /AMI *.mol`) under Windows is improved, in order to tolerate possible large expansion volumes

5

- Source code was significantly changed in order to ensure multithread execution safety of the InChI library
- Several minor changes (including refactoring)
- Updated documentation

v1.04 release 2011-10-12

Added

- Support for chemical element numbers 105-112 (see release notes for detailed information)
- Possibilty of processing mult iple input files at a single run added to inchi-executable (both on Windows and Linux versions); common file name wildcards are recognized

Changed

- Software licence changed from GNU Lesser General Public Licence (LGPL) version 2.1 to the more permissive IUPAC/InChI-Trust Licence for International Chemical Identifier (InChI) Software version 1.04, September 2011 ("IPUCA/InChI-Trust Licence No. 1.0")
- Standard valences for No, Lr, and Rf were updated
- Several minor changes in source code

Removed

- Removed support for CML input files and related source code
- Removed unnecessary dependence of the inchi-1 executable for Windows on Microsoft run-time dynamic libraries was eliminated

Fixed

- Fixed bug in the normalization procedure for some structures (containing a radical at an atom in an aromatic ring) which may result in different InChI strings for the same molecule, depending on the original order of the atomic numbers; this fix repairs the issue reported Dmitry Pavlov, SciTouch

v1.03 release 2010-06-15

Added

- New command line option `SLUUD` to distinguish stereo labels 'u' for "unknown" and '?' for "undefined"
- Experimental new line options `KET` and `15T` provide access to advanced tautomerism detection; `/Ket` (`-Ket` under Linux) for keto-enol tautomerism; `/15T` (`-15T` under Linux) for 1,5-tautomerism

- New structure perception option `NEWPSOFF` points both ends of wedge bonds to stereocenters
- New structure perception option `DoNotAddH` makes all hydrogens in input structure explicit
- New structure perception option `SNon` to ignore stereo
- New structure perception option `SRel`/`SRac` to use relative/racemic stereo
- New structure perception option `SUCF` to use chiral flag in MOL/SD file record: if On - use absolute stereo, Off - relative
- New InChI creation option `SUU` to always indicate unknown/undefined stereo
- New InChI creation option `RecMet` to include reconnected metals in results
- New InChI creation option `FixedH` to include "Fixed H layer"
- New command line option `SaveOpt` allows to append non-standard InChI string wih saved InChI creation options
- New command line option `XHash1` allows to output the res of 256-bit SHA-2 signature for 1st block
- New command line option `XHash2` allows to output the res of 256-bit SHA-2 signature for 2nd block

Changed

- For `InChI2InChI` conversion generating standard InChI strings out of standard ones is disabled; conversion to non-standard ones is allowed (see release notes for detailed information)
- Minor technical clarification is made in references to GNU Lesser General Public License, LGPL
- Source code of CML reade is published with InChI package, as free software under the same LGPL licence as InChI itself, thanks to Peter Murray-Rust and Simon (Billy) Tyrell, Unilever Centre for Molecular Sciences Informatics, University of Cambridge, UK

Fixed

- Fixed bug in normalization procedure for some structures (containing N2(+) fragment) which may result in producing different InChI strings for the same molecule, depending on original order of the atomic numbers - thanks to Timo Boehme
- Fixed bug in normalization for structures containing positively charged tetra-coordinated phosphorus (or sulfur) connected to the negatively charged oxygen - thanks to Hinnerk Rey
- Fixed bug in treating stereochemistry of allenes with non-carbon substituents - thanks to Burt Leland

v1.02 release 2008-09-15

Added

- Introduced **Standard InChI** and **Standard InChI Key** (for detailed information see release notes)
- Added command line option `InpAux` for use in combination with option `STDIO` if the input stream contains structure(s) in AuxInfo form

Fixed

- Fixed missing "fixed-H" iso segement in case of single removed D(+)
- Fixed bug at reading InChI
- Fixed "DALKE_BUGS" (see https://sourceforge.net/p/inchi/mailman/inchi-discuss/thread/028b01c77cac%24c721f8e0%248801a8c0@xempc3/ for detailed information) - Thanks to A. Dalke
- Fixed bug that leads to missed charge in some cases when /o is present
- Fixed bug of i2i conversion SAbs–>(SRel||Srac)
- Fixed bug of i2i conversion (missed empty /t)
- Fixed bug of i2i conversion (missed repeating /s in FI after F for multi-component case)
- Fixed bug that in some cases (dependent on ordering numbers), moving a charge from terminal H to heavy atom resulted in neutralizing H but not adjusting charge of heavy atom
- Fixed bug which leads for different InChI on atomic permitations for systems containing radical at atom in aromatic ring

v1.01 release 2005-04

Added

- New command line option `InChI2InChI` checks input InChI string for significatn syntax errors, may remove layers, result is an InChI string (for detailed information see release notes)
- New command line option `InChI2Struct` converts input InChI into a 0D structure (for detailed information see release notes)
- New command line option `SPXYZ` treats Phosphines as stereogenic
- New command line option `SAsXYZ` treats Arsines as stereogenic
- New command line option `FixSp3bug` or `FB` activates fixes of two known bugs in stereochemical sp3(/t) segement

Fixed

- Several bugs have been fixed, most of the bugs were discussed at InChI-discuss (https://lists.sourceforge.net/lists/listinfo/inchi-discuss)
- Fixed bug that a stereocenter connected by 3 bonds in a 2D structure could be undected if an average bind length is greater than 20

- Fixed bug that a parity of a stereocenter connected by 4 bonds in a 2D structure such that a stereobond exactly overlaps with another single bond may become "undefined" depending on the order of the atoms in the structure